AttributeInMetadataConfiguration

Overview

The `AttributeInMetadata` type is a Matcher which filters results based on `<md:RequestedAttribute>` elements within the request-indicated `<md:AttributeConsumingService>` in the SP's metadata. The parameterization controls

- Whether the `<md:RequestedAttribute>` naming is applied directly or inferred from a rudimentary algorithm based on the IdP's attribute encoding rules (`attributeName` and `attributeNameFormat`)
- The behavior when the metadata contains no `<md:RequestedAttribute>` elements (via `matchIfMetadataSilent`)
- The behavior with respect to the `isRequired` XML attribute
- Whether this is a Matcher or a PolicyRule (via `attributeID`)

Value matching is purely string-based. Only string attribute values of the input attribute are inspected and they are compared with a string representation of each of the values in the `RequestedAttribute`. Only matching values are added to the Permit or Deny List.

> **AttributeInMetadata or MappedAttributeInMetadata?**
>
> The `AttributeInMetadata` and `MappedAttributeInMetadata` matchers have significant overlap. In practice, use `AttributeInMetadata` if you need to base the rule on the SAML attribute naming, and `MappedAttributeInMetadata` if you need to compare non-string values or are concerned about the extra costs of constantly performing the indirection between the SAML naming in the metadata and the internal encoding rules in the IdP.

Schema Type and Location

The `AttributeInMetadata` type is defined in the `urn:mace:shibboleth:2.0:afp` namespace, the schema for which can be located at http://shibboleth.net/schema/idp/shibboleth-afp.xsd

The deprecated `saml:SAMLAttributeInMetadata` type is defined in the `urn:mace:shibboleth:2.0:afp:mf:saml` namespace, the schema for which can be located at http://shibboleth.net/schema/idp/shibboleth-afp-mf-saml.xsd

Reference

Attributes

Name	Type	Default	Description
`attributeName`	String	optional	If this attribute is present, a `<md:RequestedAttribute>` element with this `Name` is searched for. If this attribute is not present, then the attribute encoders associated with the internal IdP Attribute under consideration are consulted and the first matching (based on `NameFormat` and `Name`) `<md:RequestedAttribute>` is used.
`attributeNameFormat`	String	optional	If this attribute is present, provides additional filtering of the `<md:RequestedAttribute>` information, such that it will only be matched if one of the following conditions is true: • The value of `attributeNameFormat` attribute matches the value of the `NameFormat` XML attribute *or* • The `NameFormat` XML attribute is not present *or* • The value of the `NameFormat` XML attribute is `urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified` Use of the `attributeNameFormat` attribute is meaningless if `attributeName` is not present.
`matchIfMetadataSilent`	Boolean	false	If true then all input values are returned if and only if the metadata contains no `<md:RequestedAttribute>` information.
`onlyIfRequired`	Boolean	true (erroneously false in release prior to 3.2, see here)	If this is true and the corresponding `<md:RequestedAttribute>` element does not specify `isRequired="true"` , then no values are matched.
`attributeID`	String	optional	If this attribute is present, then this is a PolicyRule returning true if the Matcher, when applied to the attribute with this ID, matches any values. See AttributeValueString for an exmaple of how attributeID changes the meaning of a Matcher in a slightly less daunting scenario.

Child Elements

None

Example 1

Suppose an SP has the following requested attributes in metadata:

```
<md:RequestedAttribute FriendlyName="eduPersonPrincipalName"
   Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri" isRequired="true"/>

<md:RequestedAttribute FriendlyName="displayName"
   Name="urn:oid:2.16.840.1.113730.3.1.241"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>

<md:RequestedAttribute FriendlyName="mail"
   Name="urn:oid:0.9.2342.19200300.100.1.3"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri" isRequired="true"/>
```

Then an IdP >= v3.2 with the following configuration will release eduPersonPrincipalName and mail as wire attributes to the above SP provided that they are configured with attribute encoders that match the SAML naming above. An IdP <v3.2 will release displayName additionally.

```
<afp:AttributeFilterPolicy id="releaseEssentialAttributesToAnySP">

  <afp:PolicyRequirementRule xsi:type="basic:ANY"/>

  <afp:AttributeRule attributeID="eduPersonPrincipalName">
    <afp:PermitValueRule xsi:type="AttributeInMetadata"/>
  </afp:AttributeRule>

  <afp:AttributeRule attributeID="displayName">
    <afp:PermitValueRule xsi:type="AttributeInMetadata"/>
  </afp:AttributeRule>

  <afp:AttributeRule attributeID="email">
    <afp:PermitValueRule xsi:type="AttributeInMetadata"/>
  </afp:AttributeRule>

</afp:AttributeFilterPolicy>
```

Example 2

Now suppose an SP has the following requested attributes in metadata:

```
<md:RequestedAttribute FriendlyName="metaSharedUserID"
   Name="http://example.org/attribute/metaSharedUserID"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>

<md:RequestedAttribute FriendlyName="metaPersonName"
   Name="http://example.org/attribute/metaPersonName"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>

<md:RequestedAttribute FriendlyName="metaEmailAddress"
   Name="http://example.org/attribute/metaEmailAddress"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
```

Then two IdPs with the following configurations will release the indicated wire attributes to the above SP:

```
<afp:AttributeFilterPolicy id="
mapAndReleaseEssentialAttributesToAnySP">

  <afp:PolicyRequirementRule xsi:type="ANY"/>

  <afp:AttributeRule attributeID="
eduPersonPrincipalName">
    <afp:PermitValueRule xsi:type="
AttributeInMetadata"
        attributeName="http://example.org/attribute
/metaSharedUserID"
        attributeNameFormat="urn:oasis:names:tc:
SAML:2.0:attrname-format:uri"/>
  </afp:AttributeRule>

  <afp:AttributeRule attributeID="displayName">
    <afp:PermitValueRule xsi:type="
AttributeInMetadata"
        attributeName="http://example.org/attribute
/metaPersonName"
        attributeNameFormat="urn:oasis:names:tc:
SAML:2.0:attrname-format:uri"/>
  </afp:AttributeRule>

  <afp:AttributeRule attributeID="email">
    <afp:PermitValueRule xsi:type="
AttributeInMetadata"
        attributeName="http://example.org/attribute
/metaEmailAddress"
        attributeNameFormat="urn:oasis:names:tc:
SAML:2.0:attrname-format:uri"/>
  </afp:AttributeRule>

</afp:AttributeFilterPolicy>
```

```
<afp:AttributeFilterPolicy id="
mapAndReleaseEssentialAttributesToAnySP">

  <afp:PolicyRequirementRule xsi:type="ANY"/>

  <afp:AttributeRule attributeID="
eduPersonUniqueId">
    <afp:PermitValueRule xsi:type="
AttributeInMetadata"
        attributeName="http://example.org/attribute
/metaSharedUserID"
        attributeNameFormat="urn:oasis:names:tc:
SAML:2.0:attrname-format:uri"/>
  </afp:AttributeRule>

  <afp:AttributeRule attributeID="givenName">
    <afp:PermitValueRule xsi:type="
AttributeInMetadata"
        attributeName="http://example.org/attribute
/metaPersonName"
        attributeNameFormat="urn:oasis:names:tc:
SAML:2.0:attrname-format:uri"/>
  </afp:AttributeRule>

  <afp:AttributeRule attributeID="email">
    <afp:PermitValueRule xsi:type="
AttributeInMetadata"
        attributeName="http://example.org/attribute
/metaEmailAddress"
        attributeNameFormat="urn:oasis:names:tc:
SAML:2.0:attrname-format:uri"/>
  </afp:AttributeRule>

</afp:AttributeFilterPolicy>
```

Note that both IdPs have an attribute release policy that relies on the same set of requested attributes, but the requested attributes are mapped to different wire attributes in each case.